To the knowledge of the submitter, none of the prior art discussed herein has been cited in the file history of the application.

To the knowledge of the submitter it is believed that all of the submitted references were published before the earliest priority date of 10/775,744, 03-04-2003.

Remarks

Claim 1 recites: *A method for monitoring time series, comprising: a means for storing data from multiple input time series; a means for executing persistent queries; a means for dynamic management of persistent queries; a means for performing online computation of statistics.*

We share below a sample set of relevant prior art embodied in academic published papers and in the public domain.

Please look at all the papers referenced within Appendix I. All of the research work is related (there is a university research community covering persistent, continuous queries over time series streaming data "data streams" as an outcrop from the RDBMS research community). Additionally, each paper has multiple references to other related research work and related prior art.

We select here one representative paper, reference "MSNC" in the table of Appendix I below: "Monitoring Streams – a new class of Data Management Applications" by Brian Babcock et al., that clearly discloses each and every element of claim 1 and therefore anticipates claim 1 under one or more sections of 35 USC 102.

Specifically, "MSNC" discloses (and is substantially similar to and based on the even earlier "CS02" paper written by (substantially) the same authors):

A method for monitoring time series	*From Section 1 Paragraph 2 Sentence 2:* Monitoring applications are applications that monitor continuous streams of data. *From Section 1 Paragraph 8:* Monitoring applications are very difficult to implement in traditional DBMSs. First, the basic computation model is wrong: DBMSs have a HADP model while monitoring applications often require a DAHP model. In addition, to store time-series information one has only two choices. *From Section 1 Paragraph 11:* In this paper, we describe a prototype system, Aurora, which is designed to better support monitoring applications. *From Section 2 Aurora System Model:*

	Paragraph 1: Aurora data is assumed to come from a variety of data sources such as computer programs that generate values at regular or irregular intervals or hardware sensors. We will use the term data source for either case. In addition, a data stream is the term we will use for the collection of data values that are presented by a data source. Each data source is assumed to have a unique source identifier and Aurora timestamps every incoming tuple to monitor the quality of service being provided.
a means for storing data from multiple input time series;	*From Section 4 Run-Time Operation:* **4.2 Storage Management** The job of the Aurora Storage Manager (ASM) is to store all tuples required by an Aurora network. There are two kinds of requirements. First, ASM must manage storage for the tuples that are being passed through an Aurora network, and secondly, it must maintain extra tuple storage that may be required at connection points. **Queue Management.** Each windowed operation requires a historical collection of tuples to be stored, equal to the size of the window.
a means for executing persistent queries;	*From Section 2 Aurora System Model:* **2.1 Operators** Aurora contains built-in support for eight primitive operations for expressing its stream processing requirements. Included among these are windowed operators that operate on sets of consecutive tuples from a stream ("windows") at a time. Every windowed operator applies an input (user-defined) function to a window and then advances the window to capture a new set of tuples before repeating the processing cycle. Slide advances a window by "sliding" it downstream by some number of tuples.
a means for dynamic management of persistent queries	*From Section 2 Aurora System Model: Subsection 2.2 Query Model: Paragraph 1:* **2.2 Query Model** Aurora supports continual queries (real-time processing), views, and ad-hoc queries all using substantially the same mechanisms. *Same subsection: Paragraph 3:*

	The dark circles on the input arcs to boxes b_1 and b_2 represent connection points. A connection point is an arc that will support dynamic modification to the network. New boxes can be added to or deleted from a connection point. When a new application connects to the network, it will often require access to the recent past. As such, a connection point has the potential for persistent storage (see Section 4.2).
a means for performing online computation of statistics	*From Section 2 Aurora System Model: Subsection 2.1 Operators: paragraph 1:* This operator could be used to perform rolling computations, as in a query that continuously determines the average value of IBM stock over the most recent three hours. Tumble resembles Slide except that consecutive windows have no tuples in common. Rather, Tumble effectively partitions a stream into disjoint windows. This is useful, for example, when calculating daily stock indexes, where every stock quote is used in exactly one index calculation. Latch resembles Tumble but can maintain internal state between window calculations. This is useful for "infinite window" calculations, such as one that maintains the maximum or average value of every stock, maintained over its lifetime.

Claim 2 recites: The method as recited in claim 1, further comprising: a means for dynamic management of windows.

"MSNC" paper clearly discloses each and every element of claim 2 and therefore anticipates claim 1 under one or more sections of 35 USC 102.

Specifically, "MSNC" discloses:

a means for dynamic management of windows;	*From Section 4 Run-Time Operation: Sub-section 4.2:* **4.2 Storage Management** The job of the Aurora Storage Manager (ASM) is to store all tuples required by an Aurora network. There are two kinds of requirements. *Paragraph 2:* **Queue Management.** Each windowed operation requires a historical collection of tuples to be stored, equal to the size of the window. Moreover, if the network is currently saturated, then additional tuples

	may accumulate at various places in the network. As such, ASM must manage a collection of variable length queues of tuples.

Claim 3 recites: The method as recited in claim 1, further comprising: a means for using historical values in present windows to help populate inserted windows.

"MSNC" clearly discloses each and every element of claim 3 and therefore anticipates claim 1 under one or more sections of 35 USC 102.

Specifically, "MSNC" discloses (boxes represent persistent queries or components thereof):

a means for using historical values in present windows to help populate inserted windows.	*From Section 2.2 query Model : Para 3: (boxes represent queries, here adding new queries and associated new windows):* The dark circles on the input arcs to boxes b_1 and b_2 represent connection points. A connection point is an arc that will support dynamic modification to the network. New boxes can be added to or deleted from a connection point. When a new application connects to the network, it will often require access to the recent past. As such, a connection point has the potential for persistent storage (see Section 4.2). Persistent storage retains data items beyond their processing by a particular box. In other words, as items flow past a connection point, they are cached in a persistent store for some period of time. They are not drained from the network by applications. Instead, a persistence specification indicates exactly how long the items are kept. In the figure, the left-most connection point is specified to be available for two hours. This indicates that the beginning of time for newly connected applications will be two hours in the past. *From Section 4 Run-Time Operation: Sub-section 4.2 (support for ad-hoc queries):* **Connection Point Management.** As noted earlier, the Aurora application designer indicates a collection of connection points, to which collections of boxes can be subsequently connected. This satisfies the Aurora requirement to support ad-hoc queries. Associated with each connection point is a history requirement and an optional storage key. The history requirement indicates the amount of historical information that must be retained. Sometimes, the amount of retained history is less than the maximum window size of the successor

	boxes. In this case, no extra storage need be allocated. The usual case is that additional history is requested.

Conclusion
In view of the above reasoning, it is clear that "MSNC" clearly discloses each and every element of claims 1-3, and therefore anticipates claims 1-3 under one or more sections of 35 USC 102.

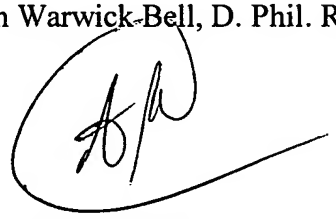
The submitter respectfully requests that these references and arguments be made of record in this case.

This paper is being submitted simultaneously to:

Eric Bax
iSpheres Corporation
640 Third St.
Oakland CA 94607
UNITED STATES (US)

Respectfully submitted,

Adam Warwick Bell, D. Phil. Reg. No. 43,490



Bell & Associates (Customer No. 039843)
416 Funston Avenue Suite 100,
San Francisco CA 94118